Storage Manager at MTCC

Lessons Learned and Development

Harry Cheung, Fermilab

- **Lessons Learned from MTCC1**
  - Directly writing POOL/ROOT files is too slow
    - ROOT de-serialization is a fundamental limit
  - Streamer files are too large and ratio of streamer file size to “actual data” size was too variable
    - Serialization of objects lost the compression contained in ROOT
  - Insufficient resources to convert streamer file to root files in real-time
  - Dual configuration file feature of SM not desirable
  - The SM needed to be more stable and fault tolerant
  - Need to ensure streamer files are really used in an intermediate stage only - the format is designed to be temporary and should not be stored in MSS
    - Issues with de-serialization of old objects in a newer CMSSW version
  - Need better feedback and communication with MTCC people, and better deployment method (last but most important!)

To Implement for MTCC2

- **Proposed changes for MTCC2**
  - **Already for end of MTCC1:** Christoph Paus and his strong MIT group to work on SM operations, deployment, and to include SM development, particularly for operations-related and interaction with Tier-0
  - Use compression in streamer files
  - Deploy support for multiple event consumers
  - Convert streamer files to Pool/Root files in real-time(?)
  - Increase 10K events/file limit
- **Current development planned to use in MTCC2 (depends on time)**
  - More thread-safe and fault tolerant code
  - Filling of trigger path status bits in Event Header
  - Ability to write multiple streamer files based on trigger path status bit selection
  - Ability for event consumers to get events based on trigger path status bit selection
  - Ability for cmsRun converter to read in multiple streamer files
  - Entry in a statistics file (“ascii text DB”) for each streamer file written
  - Other items from Christoph’s team(?)